An Application of Neural Networks to Sequence Analysis and Genre Identification

David Bisant, Ph.D.
The Laboratory for Physical Sciences
8050 Greenmead Drive ,College Park, MD 20740
bisant@umbc.edu

Abstract

This study borrowed sequence analysis techniques from the genetic sciences and applied them to a similar problem in email filtering and web searching. Genre identification is the process of determining the type or family of a given document. For example, is the document a letter, a news story, a horoscope, a joke, or an advertisement. Genre identification allows a computer user to further filter email and web sites in a way that is totally different than topic-based methods. This study presents original research in an application of neural networks to the genre identification problem.

The data for the study came from a database constructed by the author and his colleagues. The data consisted of descriptive features and the genre classification, as judged by a human, from over 5000 different documents. Ten different genres were represented. The descriptive features consisted of 89 different measurements of each document such as average word length, the number of numeric terms, the proportion of present tense verbs, etc. The data was divided into 2 sets, with 75% for training and 25% for testing.

The first neural network applied was a very basic single layer network that achieved 79% correct classifications on the testing data. This performance was equivalent to the previous best method on the problem, decision trees. When more complex neural networks were applied to the problem, performance increased significantly. The best performance of 86% correct classifications was achieved by a network with a single hidden layer of 300 units. Increasing the number of hidden layers, or changing the number of hidden units did not improve performance. A weight decay process also did not improve performance.

The analysis of the features indicated that 2nd order information was being exploited by the networks for better performance. This means that neural networks will outperform statistical models or other methods that only utilize 1st order information.

Introduction:

Genre Identification:

The goal of email and web site filtering is to eliminate unwanted documents from distracting a user's attention.. With the exponential increase in email traffic volume, filtering is becoming increasingly important to maintain a user's productivity. Currently, these methods can filter and select based on sender recognition, date information, particular topical combinations of dictionary keywords, etc (Sahami et al, 1998). The success of these methods has been limited. Recent research in topic selection has hoped to improve this, but even if an email or web site is on topic, it can still be of useless value. For example, an advertisement for a U.S. mutual fund may have little value when conducting an academic study, but may be of the same topic as the U.S. central bank investment plans. On the other hand, if an academic wishes to determine how the commercial sector is reacting to government economic activities, they may be interested in financial advertisements.

Genre identification seeks to determine the family or type of a document or email. Some of the families include emails from irritating acquaintances, advertisements, news stories, horoscopes, internet chat, and many others. Some 40 to 50 different families or genres have been identified. So far research in my organization has focused on the 10 different genres identified below:

1. Advertisement
2. Bus. Correspondence
3. Data Entry Forms
4. E-Mail Admin
5. E-zine
6. Friend Correspondence
7. Internet Chat
8. News Service
9. Notices
10. Technical Data

Genre identification is orthogonal to topic spotting, but much like topic spotting, it is an easy process for humans. When dealing with large volumes of data, however, the limitations of human processing becomes a problem. Research at my organization and elsewhere (Kessler, Nunberg, and Schutze, 1997; Stamatatos, Fakotakis, and Kokkinakis, 2001) has sought to identify genres automatically. Once the genre is identified automatically, this information could be kept with desired documents as a metatag to assist in answering further queries or for further processing by summarization or filing algorithms.

Before a document can be classified by an automated system, several measurements have to be taken; such as the number of words, their average length, the presence of specific characters and words, etc. These measurements, or "features", form a feature vector, which is then used by the

system to perform the actual classification. Since the system has to recognize the pattern of features appropriate for each class, the genre identification problem is considered a pattern classification problem. Several machine learning techniques have been effectively applied to pattern classification problems. Early research on the problem (Biber, 1993) was limited to very few genres or to the binary problem of sorting junk email from desired email (Sahami et al, 1998). Later work on the problem (2, 3) focused on a larger scale and tried respectable machine learning techniques such as Support Vector Machines (SVM)(DoD 2000), C4.5 (DoD 2001) , and neural networks or statistical methods (Kessler, Nunberg, and Schutze, 1997; Stamatatos, Fakotakis, and Kokkinakis, 2001) with good results. The goal of this study was to improve performance in automated genre identification using neural network techniques, originally developed by David Rumelhart and others (Rumelhart, Hinton, and Williams, 1986). The neural network models were effective at accomplishing this. The neural network methods also revealed some interesting information in how they were classifying the data.

Methods:

Data Preparation

The UM / DoD Laboratory for Physical Sciences has created a repository of emails and web pages from open and internal sources and has generated programs for deriving the measurements or features of each one. One of the repositories covers ten different genres and contains over 5000 different types of documents, derived from public data sets and internal sources. The author was provided with a large set of feature vectors from this repository, one for each document, and an index describing the genre classification for each document. Each feature vector had been generated by the feature extractor which creates 89 different features for a document, most of which were originally conceived by early pioneers in the field. They consist of such measurements as average word length, how many words of a certain type appear in the document, character measurements, etc. A subset of these features are listed in Appendix A. All the features come normalized to a value between 0 and 1. Using information from the classification index, an output vector was created for each document. The output vector consisted of ten numbers, nine of which were set to 0 while one number, corresponding to the type of genre for the document, was set to 1. The feature vectors were combined with the output vectors to produce a set of input/output pairs. An example input/output pair is illustrated in Figure 1. The set of pairs was randomly divided into a training set (4360 pairs) and a testing set (1454 pairs). The testing set was reserved for evaluation of the networks only

Training the Model

The neural network simulator used for the project was UMNET developed at Stanford University and the University of Maryland. A variety of different network architectures were tried on the problem. All of them had 89 inputs and 10 outputs and all the layers were fully connected. Networks were tried with 0, 1, and 2 hidden layers. Regardless of the architecture, a typical training run started by randomizing the weights in the network to small random values. All training patterns were then selected randomly and presented to the network. Back-propagation was used as the training algorithm. Usually 150,000 to 250,000 training patterns were necessary for convergence. The network was stopped and tested periodically (e.g. every 20,000 iterations) on the test set. A record of performance was recorded as training progressed on both the test set and the training set to produce a training curve, a graph of network performance during training. Some exploration was done with various training parameters such as the learning rate.

Results:

All the training curves presented below are graphs of the true classification error (CE=incorrect /total). Although the sum of squared error (SSE) was used to optimize the networks, the CE is a more relevant measure of classification performance. Figure 2 shows the training curves for a network with 89 fully connected input units and 10 output units. This is a very basic network with only a single layer of adjustable weights. Its peak performance on the test set, in the range of 79% correct classifications was identical to

0.903 0.051 0.046 0.006 0.400 0.400 0.000 0.200 0.125 0.375 0.000 0.375 0.125 0.000 0.000 0.000 0.000 0.034 0.017 0.333 0.000 0.667 0.063 0.000 0.023 0.051 0.046 0.017 0.000 0.156 0.917 0.441 0.175 0.097 0.866 0.623 0.012 0.375 0.000 0.300 0.353 0.000 0.118 0.025 1.000 1.000 1.000 1.000 1.000 0.100 1.000 1.000 1.000 1.000 0.137 1.000 0.167 1.000 1.000 1.000 1.000 0.500 1.000 1.000 1.000 1.000 1.000 1.000 0.500 0.049 0.951 0.000 0.001 0.305 0.008 0.003 0.200 0.500 0.333 0.500 1.000 0.500 1.000 0.765 1.000 1.000 0.628 0.500 1.000

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0

Figure 1: An example input/output pair used in this study. This one corresponded to a document from the Notice genre.

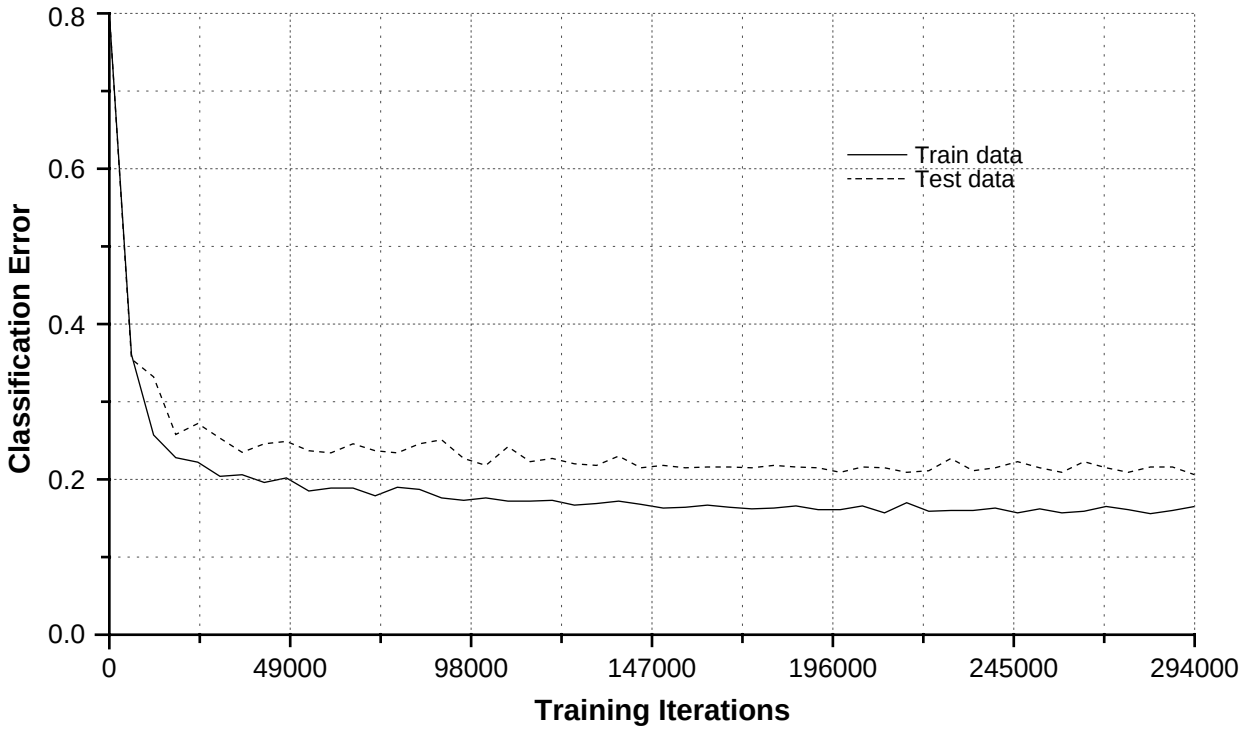

Figure 2: The training graph for a single layer network with 89 inputs and 10 outputs. The Y axis is the classification error as defined earlier. The X axis represents the number of patterns that have been used to train the network. Results are reported from the test data

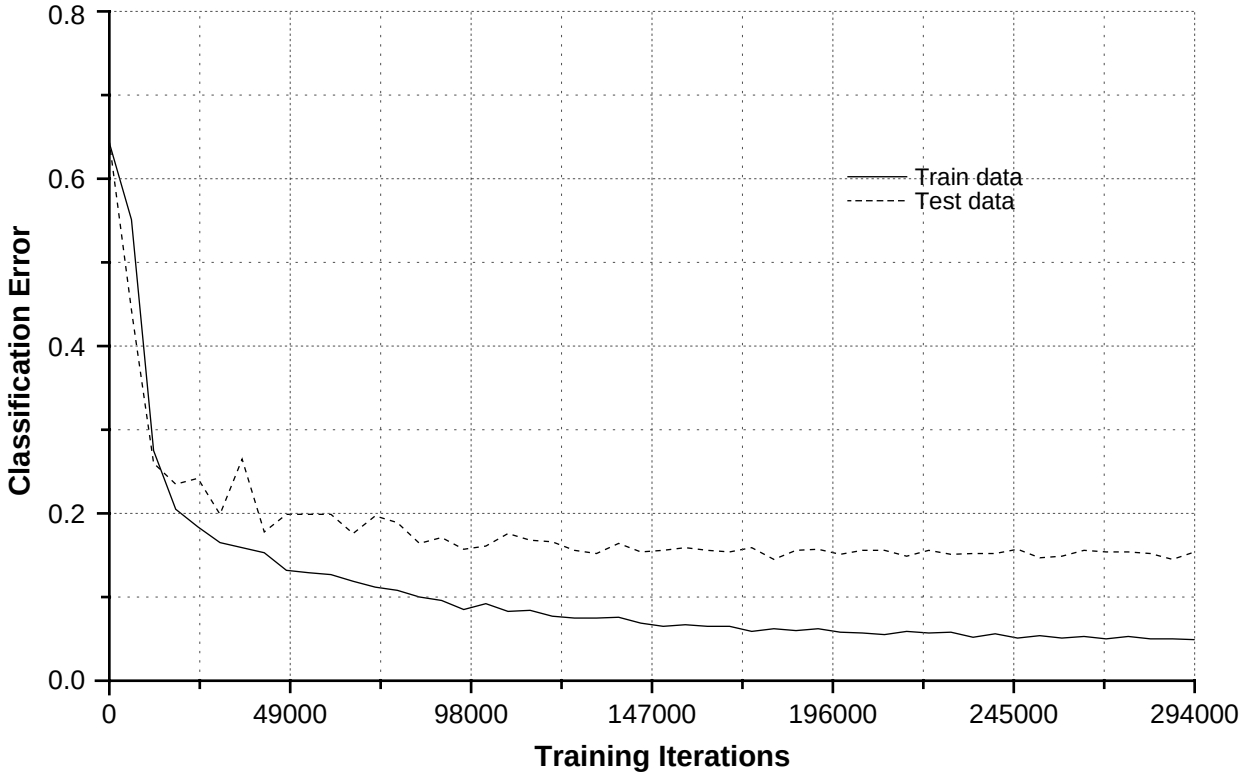

Figure 3: The training curves for a multilayer network with 89 input units, 300 units in the hidden layer, and 10 output units (89-300-10). The layers are fully connected.

.

the performance of a decision tree. By adding a single layerof 50 hidden units to the network, performance increased up to 82%. It was also noticed that the testing performance improved as the training progressed, much as it did for the 89-10 network in Figure 2. The fact that performance continued to improve on the test data as training progressed indicated that the networks could be enlarged without fear of over-fitting.

Enlarging the hidden layer to 300 improved performance and still did not demonstrate over-fitting, see Figure 3. The peak performance was 86% correct classifications. This particular network took 60 seconds to train on the full set of training data (4360 pairs) on a Sun ULTRA 60. This corresponds to 67 minutes for a full training run of 294,000 pairs. The testing time is much faster since no weight adjustments take place. A full pass through the testing set (1454 pairs) took less than 3 seconds on the same machine. Figure 4 illustrates the performance of the network for each specific genre classification. The error curves are separated to avoid confusion by adding the category number to the error for each category. The network does not perform as well on some categories as others. Category 9 (notices) is a difficult category for the network. It also has difficulty with 2 (business correspondence), 5 (E-zine), and 8 (News Services), though to a lesser degree. A close examination indicated that some of the curves were inversely correlated.

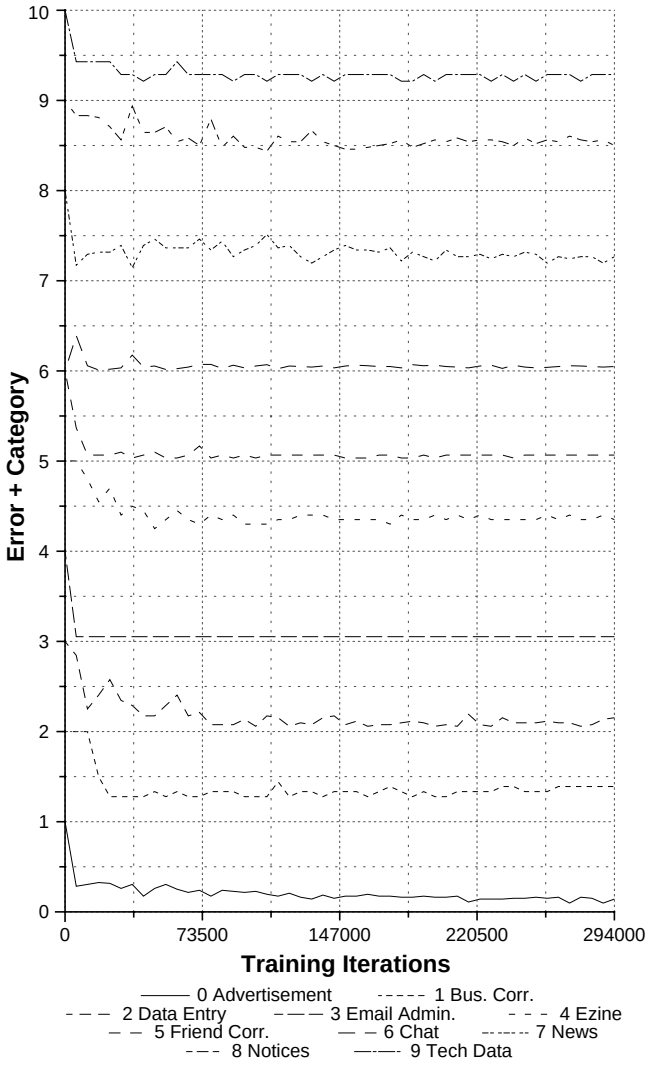

Figure 4: Performance error on each specific genre by the 89-300-10 network.

The inverse correlation's were more pronounced for the smaller networks. Most of the performance curves had an inverse correlation between categories 3 (Data Entry Form) and 9 (Notices), or between category 8 (News) and 9. This indicates that the networks are confusing these categories. To maximize performance on one of these categories, the network would have to minimize performance on the other. This could be the result of improperly prepared data, limited features, or just two categories that tend to be very similar.

Adding a weight decay term (Weigend, Rummelhart, and Huberman, 1990) to the objective function, adding additional hidden layers, or increasing the size of the hidden layer over 300 units did not improve performance. It was interesting that even the larger networks did not demonstrate evidence of overfitting.

Comparison to Decision Trees and Other Methods:

One of the groundbreaking studies on this problem (DoD 2000) applied SVM to the classification of 4 genres with a success rate of 86%. The researchers estimated that SVM could perform in the 80-84% range on the ten-genre problem. Results were available for only one other method on the ten-genre classification problem, a decision tree approach which was applied at my organization. This approach produced 79% correct classifications overall. On a category by category basis, a comparison of the neural network versus the decision tree is presented in Table 1. For both methods, the performance for a given category was derived by the following formula then expressed as a percentage:

$$1 - \frac{\text{False Negatives}}{\text{True Positives} + \text{False Negatives}}$$

A true positive for a given category is a document belonging to that category which was correctly classified by the network. A false negative for a given category is a document belonging to that category which was incorrectly classified by the network as belonging to a different category.

The 89-300-10 neural network outperformed the decision tree on nine out of the ten categories. It is interesting to note that there are significant differences in the distribution of the scores between the two methods, indicating that the methods derive their classifications in different ways, Perhaps this difference could be exploited by combining the

.

Genre	Decision Tree	89-10 Net	89-300-10 Net
Advertisement	**79**	**64**	**87**
Correspondence	**64**	**67**	**67**
Data Entry	**64**	**68**	**89**
Email	**94**	**94**	**95**
Ezine	**62**	**66**	**70**
Inbox	**84**	**92**	**93**
Internet Chat	**91**	**91**	**96**
News Service	**65**	**66**	**78**
Notices	**58**	**49**	**51**
Tech Data	**57**	**70**	**71**

Table 1: Performance of decision trees and the neural networks for each type of genre.

methods into an ensemble configuration. A close examination of the results suggested some reason why the neural network performed better than the decision tree. The 89-10 neural network has a single layer of adjustable weights and scored 79% overall. Without a hidden layer, this network can only exploit 1st order information from the features presented to it. Notice that this score is identical to the 79% score produced by the decision tree which also uses, primarily, 1st order information. The correlation between the scores was 0.85. Second order information, which is derived by combining features, can be exploited by neural networks with hidden layers. Hence, the better performance by neural networks with hidden layers indicates that there is 2nd order information which can be exploited in the problem if the correct approach is employed. This subject was further examined by an initial observation of weight magnitudes which indicated that the networks with multiple layers were reinforcing weights from a number of features which were not being similarly treated by the networks with only a single layer.

Discussion:

This paper presented an exploratory study in applying neural networks to the genre identification problem. The neural network approach provided a significant increase in performance versus the next best method, decision trees. The neural network architecture which worked the best had a single hidden layer with an 89-300-10 structure. This network was able to achieve 86% correct classifications on the data versus 79% for the decision tree. On a category by category basis, the neural network approach performed better than the decision tree on 9 out of 10 categories. A number of other neural network architectures were tried on the problem. These included networks with 0, 1, and 2 hidden layers. None of these other architectures outperformed the 89-300-10 network. A smaller or larger hidden layer also failed to provide further improvement. Adding a weight decay term to the objective function was helpful in studying feature influence, but did not improve the classification performance of the network. This was an indication that the 89-300-10 architecture is near optimal for this particular problem.

It would also be possible to train ten different networks, each of which only recognized a single genre. This was done for the study which applied decision trees. From an academic perspective this may provide some interesting information, but in actual application the effect is unknown. If a classification were to be done for a given document then the ten different networks will each make a prediction. A way would have to be found to arbitrate and select a winner. It is unknown if a system which selects a winner from ten different classifiers would perform better than a single classifier of the same type optimized to make the single straightforward classification in the first place.

The distributions of performance results on a category-by-category basis were different between the neural network and the decision tree. An ensemble of classifiers such

as neural networks, decision trees, Bayesian methods, and SVM could exploit these differences to effect improved performance. Perhaps another candidate for an ensemble would be a neural network trained with an objective function significantly different then the SSE. One possible candidate would be the cross entropy measurement. A network trained using the cross entropy method would not be expected to perform better than one trained using SSE, but might be complementary in an ensemble architecture.

Future genre identification systems will most likely have many more categories than the 10 explored in this study and it is also possible that the categories could be hierarchically structured. Past experience indicates that performance would decrease if more genre categories are added to the problem. A rough rule of thumb might be 0.5 to 1% for each category added. On the present problem, the neural networks tended to confuse the genres of Data Entry Form and Notices, or News and Notices. It is possible that additional features might be necessary to accurately discern the differences between these categories.

Improved performance on the ten genre identification problem is unlikely to come from further explorations using neural networks. Further improvement will likely come from refinement of the input features and the development of new ones. A different representation of the input features could improve performance as the current representation is optimal for the SVM learning algorithm. While some features should be represented as proportions and inverses, others might contribute more if they are not normalized in this manner.

For the problem at hand, neural networks have demonstrated effective performance at genre identification and would likely contribute to an effective solution of the problem as it expands in the future.

Acknowledgements:

The author would like to acknowledge the contributions of many colleagues at the University of Maryland and the Department of Defense in the preparation of the genre data, providing technical comments, and reviewing the manuscript.

References:

Biber, D. "The Multi-Dimensional Approach to Linguistic Analyses of Genre Variation: An Overview of Methodology and Findings." *Computers and the Humanities*. Vol 26, pp 331-345. 1993.

DoD 2000. *Support Vector Machines for Genre Identification*. DoD internal report. 2000.

DoD 2001. *Automatic Genre Identification for English Text Documents*. DoD internal report TR-R52-001-01. (forthcoming).

Kessler, B. G. and Nunberg, H. S. "Automatic Detection of Text Genre." *Proceedings of ACL. EACL-97*, pp 32-39. 1997.

Rumelhart, D. E., Hinton, G. E., Williams, R. J. *Learning representations by back-propagating errors*. Nature 323 (1986). pp 533-536.

Sahami, M. S. and Dumais, D. H. and Horvitz, E. "A Bayesian Approach to Filtering Junk E-Mail." *Learning for Text Categorization: Papers from the 1998 Workshop* . AAAI Technical Report WS-98-05. 1998.

Stamatatos, E. and Fakotakis, N. and Kokkinakis, G. "Automatic Text Categorization in Terms of Genre and Author." *Computational Linguistics*. Vol 26, No 4, pp 471-495. 2001.

Weigend, A. S., Rumelhart, D. E., Huberman, B. A. "Back-Propagation, Weight-Elimination and Time Series Prediction". *1990 Connectionists Models Summer School*. (1990) pp 105-116.

Yang, Y. and Pedersen, J. "A Comparative Study on Feature Selection in Text Categorization." *Machine Learning: Proceedings of the Fourteenth International Conference 1997*. pp 412-420. 1997.

Werbos, P. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph. D. Thesis, Harvard University. (1974).

Hertz, J., Krogh, A., Palmer, R. G. Introduction To the Theory of Neural Computation. Addison-Wesley, Redwood City, California (1991).

Appendix A:

Example Features;

proportion of alpha-numeric words
proportion of numeric terms (words)
proportion of query words
proportion of past tense verbs
proportion of present tense verbs
proportion of future tense verbs
proportion of infinitive tense verbs
proportion of past to past verb tense transitions
prop. of past to present verb tense transitions
proportion of past to future verb tense transitions
proportion of present to past verb tense transitions
proportion of present to present verb tense transitions
proportion of present to future verb tense transitions
inverse of average word length
estimated standard deviation of word length
inverse average number of words per line
standard deviation of words per line
inverse of average words per sentence